Justifying IMS Performance Tuning Tools

Janet LeBlanc, IBM IMS Tools Technical Specialist

e business on demand software

Disclaimer

- IMS & IMS Tools Performance Studies results contained in this document were obtained in a controlled lab environment, therefore, the results that can be obtained in other operating environment might vary significantly. Users of this document should verify the applicability of data for their specific environment .

Is Performance Tuning a Luxury or a Necessity?

Performance

Definition - A performance problem is generally noted as bad or erratic response times or an unacceptable amount of resource usage

What causes us to investigate Performance

- Service level objectives not being met
- Users complaining about slow response
- Unexpected changes in response times or resource utilizations
- z/OS operating system showing signs of stress
- The throughput on the system is erratic
- Changes in workload which were not anticipated
- Changes in the profile of transactions

What affects Performance?

- z/OS Tuning
- IMS/TM
- IMS/DM
- Application Programming
- Application environmentals
- Hardware
  - Network

- z/OS Systems
- IMS Systems Programmers
- DBAs
- Application Programmers
  - DBAs
  - Maintenance Programmers

Why is Performance Tuning at the Application or Database Level not happening anymore?

- Application Static – no changes
- Meeting Service Level Agreement
- Performance is OK
- No money for special performance tools
- No time to do performance tuning
- No skills
- Major re-writes not possible
  - No one understands the applications anymore
- If everything is running – who cares

Why should they care?

Why should we still do application tuning?

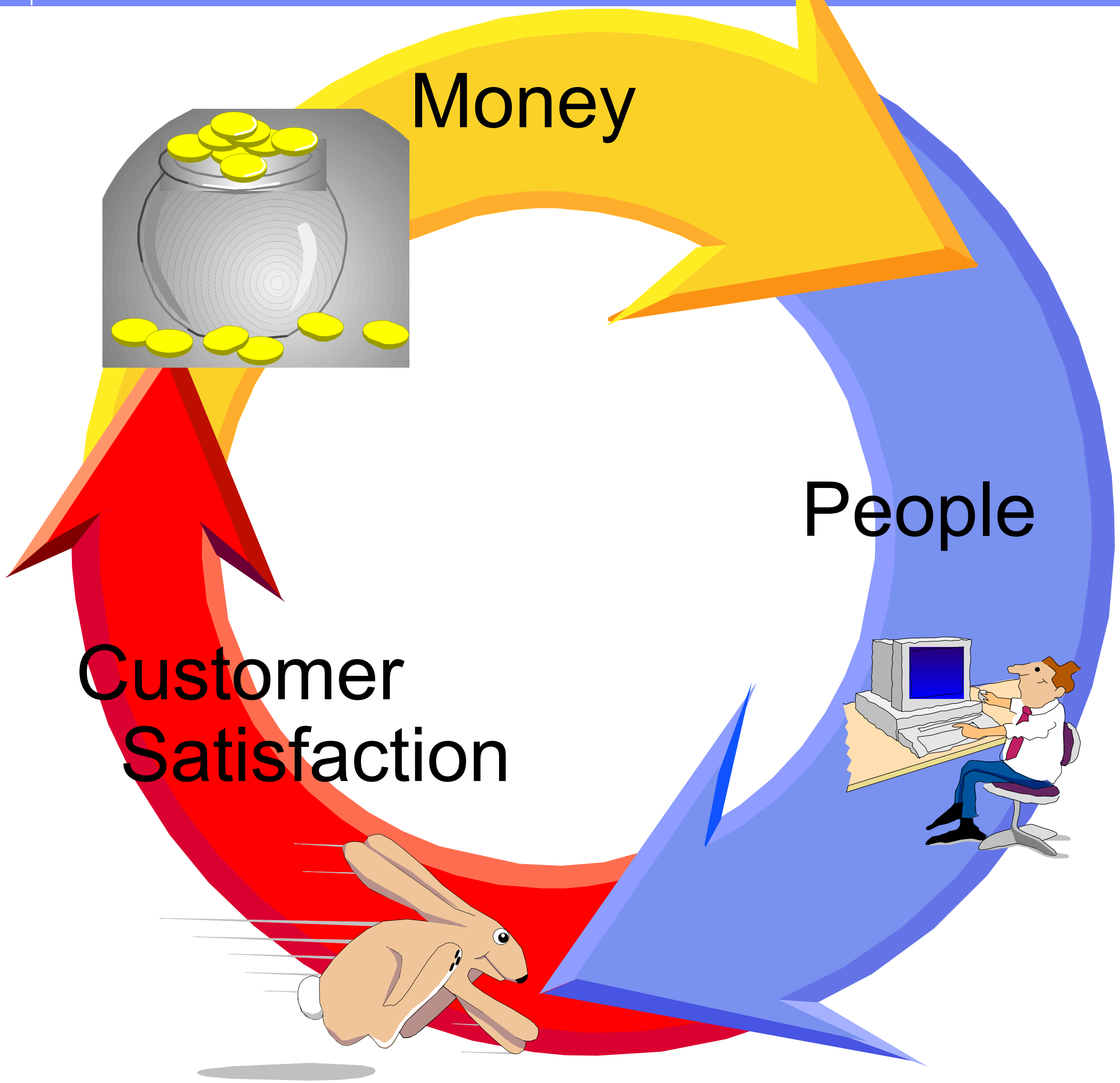
- Change is still happening:
  - New software releases affecting applications
  - Transaction Volumes are increasing
  - Data Volumes are increasing
  - Web enablement of transactions
    - IMS Connect stresses
- Reductions of 20% CPU usage/ Elapsed time is good at anytime
  - You really want to buy more hardware?
- Applications were not that well written to begin with
- Application and Database design decisions based on older releases of IMS
  - Performance Recommendations have changed

One small change can deliver
18-40% CPU time improvement

OK – now the next BIG problem

- I don't have the tools I need
- I don't have the time

- No Management Support


Money
People
Customer
Satisfaction

No Management Support

- Bottom Line
  - MONEY
  - Elapsed Time
- Get Capacity Planner onside
  - Ask to see his projection charts
  - Are you close to needed a hardware upgrade?
    - What if you can help delay it for a year?
- Who pays the bills?
- How are the bills paid?
  - Real Money?
  - Internal Money transfer?, Pseudo Money?
- What is the basis for the bill?
  - CPU time?
  - EXCPs

Step 1 – SHOW THEN THE MONEY

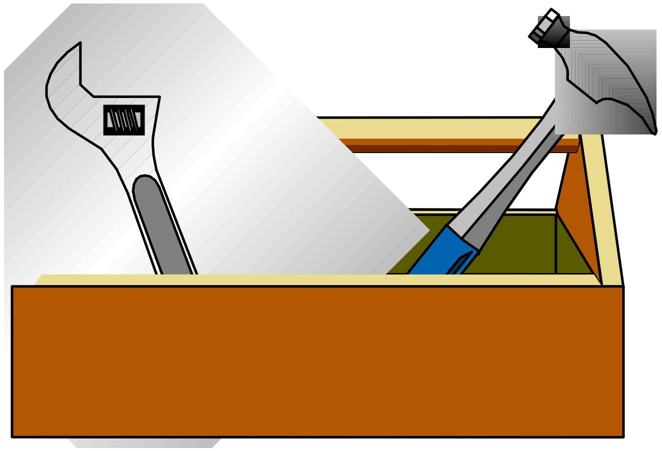
- Know your organization
- Get Sponsors
  - Not just up the chain
- Parallel Organizations
- Users
- Capacity Planners
- ….

So how do I find the money

- Where are my performance opportunities
  - What is the potential improvement
    - CPU time
    - Elapsed time
    - Processes not needed
- Track before/after statistics to prove value of performance tuning
- Proof of Concept of Performance Tuning to justify time and expense

OK but now you need time & tools

- Can it be done without tools?
  - YES but…. Needs lots of time and lots of people
  - How good is your HEX reading skills?



- Tools buy time and accuracy
- But I have a monitor!

STEP 2 – Justifying buying tools for a static environment

- Circle back – BACK to STEP 1
  - Everytime there is a stall – go to STEP 1

IMS Monitoring Tools

- IMS
  - IMS Monitor
  - File Select and Formatting Print Utility
    - DFSERA10
  - Log Transaction Analysis Utility
    - DFSILTA0
  - Knowledge Based Log Analysis (KBLA)
- Tools:
  - IMS Performance Analyzer
  - Tivoli OMEGAMON XE for IMS on z/OS
  - IMS Connect Extensions
  - IMS Problem Investigator
  - IMS Buffer Pool Analyzer
  - IMS HP Pointer Checker

Monitoring Methodology

Performance Preliminaries …Defining a Service Level Agreement

- Performance objectives must be defined as part of an service level agreement (SLA) with the relevant business unit.

- The SLA must define the following:
  - Acceptable response times to the business
  - Expected current volumes of transactions
  - Growth strategy and anticipated future volumes
  - Details of transactions and their usage

Performance Preliminaries …Defining Transaction Profiles

- A transaction profile typically covers the following:
  - Host response times:
    - Input queue time measurement
    - Total elapsed time measurement
  - The CPU time required to process the transaction
  - The number of database (DL/I and SQL) calls performed by the transaction
  - The type of database calls performed:
    - By database or table listing each database or table and the type of call
  - Number of I/Os required to perform this transaction

Performance Preliminaries
…Tracking and Trending

- Track and Trend Workload
- Understand Future Capacity Requirements
  - Capacity Planning
- Full-Time Performance Expert

Step 1 – Evaluation Process

1. Define reasonable, measurable objectives.
2. Examine the application design performance objectives and measure the application.
3. Have the objectives been met?
   a. Yes, document the status quo.
   b. No, go on to the next step.
4. Identify problem areas.
5. More information required?
   a. Yes, select appropriate tool.
   b. No, go on to the next step.
6. Take corrective actions.
7. Re-measure the application.
8. Go back to step 3 above.

- The performance objectives can be either throughput-oriented or response time-oriented. But, they must be very precise.

What do I need?

- Baseline Statistics & Historical Statistics
  - Baseline can be different things depending on what you are trying to do
    - Peak load
    - Quiet times
    - Above just after a database reorg
    - Before/After Performance Tuning

- Current Statistics
- Application Details
- Database Definition

IMS Performance Analyzer

- Daily monitoring
  - Dashboard
  - Transit response time reports
  - Management exception report
- Performance problem? Drill down with more reports!
  - Bad response time? Transit reports
  - IMS resource constraint? Resource Utilization reports
- Long-term capacity planning and service levels
  - Transaction History File – daily transaction performance
  - Load into DB2 to build a Performance Database
  - Report on host or workstation using your favorite SQL reporting tool

IMS V10 - The 56FA log record

- One record per transaction rather than per schedule (type 07)
- Additional information including:
  - OSAM and VSAM read and write counts
  - Database IO counts and elapsed times
  - Database lock elapsed times
  - External subsystem call counts
  - UOR elapsed and CPU times

Form-based reporting

- Summarize transaction activity based on any criteria, for example Region Type
- Statistical functions include average and peak percentile (to measure SLA adherence)

Summarization by Region Type

Transaction Dashboard

DASH Printed at 14:34:54 05May2006 Data from 16.03.39 29Dec2005 to 16.17.33 29Dec2005

Reg Typ	Tran Count	Avg InputQ Time	Avg Process Time	Avg OutputQ Time	Avg Total Time	Avg IMS Resp Time	Avg CPU Time	90% InputQ Time	90% Process Time	90% OutputQ Time	90% Total Time	90% IMS Resp Time	90% CPU Time
BMP	287	200	409	0	607	0	0	819	1282	0	1543	0	0
DBC	1	0	5	0	5	0	0	0	5	0	5	0	0
MPP	47017	118	63	8	189	183	18	3550	982	26	3827	3908	227
MSC	204	0	97	14	111	111	0	0	267	35	280	280	0

Transit time breakdown (averages)

Transit time breakdown (90% SLA)

Transaction volume for the day

CPU time

Design a report to meet your needs

Event collection

IMS PA V3: Transit analysis report

Total response time

Input

IMS internal response time

Output

IMS Performance Analyzer 3.3
IMS Connect Transit Analysis – HWSPROD1

From 21Apr2004 16.30.00.05 To 21Apr2004 16.40.00.09 Page 6

TranCode	Message Count		Response Time	Input Pre-OTMA	Input READ Sock	Input READ Ex	Input SAF	Process OTMA	Output Confirm	Output Post-OTMA	Output XMIT Ex	Rate /Sec	Time Outs	NAK
BALANCE	197	Avg	278.488	257.268	252.434	4.496	0.000	19.520	0.000	1.699	0.119	1	2	0
		80%	278.488	257.268	252.434	4.496	0.000	19.520	0.000	1.699	0.119			
DEPOSIT	67	Avg	247.674	226.997	222.267	4.395	0.000	18.934	0.000	1.742	0.118	1	0	0
		80%	247.674	226.997	222.267	4.395	0.000	18.934	0.000	1.742	0.118			
LOGON	123	Avg	388.525	314.452	297.292	7.614	8.853	67.305	0.000	6.767	0.162	1	0	21
		80%	392.847	319.171	302.458	7.614	9.005	81.317	0.000	11.736	0.203			
ORDER	265	Avg	357.142	275.655	262.408	6.964	5.809	73.930	0.000	7.556	0.144	1	3	0
		80%	457.589	310.779	297.228	7.556	6.343	152.839	0.000	13.602	0.201			
WITHDRAW	85	Avg	273.333	249.026	238.302	5.147	5.130	21.257	0.000	3.049	0.117	1	1	0
		80%	285.266	260.648	249.320	6.316	6.514	22.614	0.000	4.277	0.124			
Total	737	Avg	594.986	344.789	339.349	0.438	0.131	182.667	15.630	51.899	0.180	1	1	7
		80%	997.382	535.249	527.124	1.544	0.905	387.900	221.726	141.791	0.928			

Summary report – transaction activity statistically summarized

```
                                                   IMS Performance Analyzer
                                                 Transaction Transit Summary

SUMM0001 Printed at 15:16:04 07Feb2007     Data from 09.16.38 07Feb2007 to 09.43.39 07Feb2007

                           Avg         Avg       Avg        Avg        Avg        Avg
             Tran       InputQ     Process       CPU   Total IO      DB IO    DB Lock
Trancode    Count         Time        Time      Time      Count       Time       Time
IVTNO          52      521.346     295.452     1.158          0      2.056      0.000
IVTNV          36      254.697     685.690     1.555        237    217.394      0.000
Total         116      516.315     302.815     1.165          5      6.119      0.000
```

```
     Avg          Avg          Avg          Avg          Avg
OSAMRead     OSAMWrit     VSAMRead     VSAMWrit     ESAFcall
   Count        Count        Count        Count        Count
       0            0            0            0            6
       0            0            3          234            8
       0            0            0            4            7
```

V10 allows microsecond precision

Performance TID-BITs

IMS On Demand Performance

- TCP/IP outperforms SNA

Variables measured	SNA	TCP/IP
Number of connections	60000	60000 across 50 ports
CPU busy	89%	85.2%
External throughput rate (ETR)	1 829 transactions per second	1 996 transactions per second
Internal throughput rate (ITR)	2 055 transactions per second	2 342 transactions per second
Storage above 16 MB line	146 MB	119 MB

IMS Connect Extensions
IMS Performance Analyzer
IMS Problem Investigator

Event collection

IMS PA V4 : Combined Transit analysis report

IMS Performance Analyzer 4.1

Combined IMS and Connect Summary

COMBSUMM Printed at 11:15:00 04May2006 Data from 11.16.33 18Apr2006 to 12.04.17 18Apr2006

Connect Input

OTMA response time

IMS internal response time

Trancode	Tran Count	CON Avg Resp Time	Avg Input Time	Avg ReadSock In Time	Avg ReadExit Time	SAF Avg Call Time	Avg OTMAproc Time	IMS Avg Resp Time	. . .
BALANCE	14	57	0	0	0	0	51	28	
DEPOSIT	432	540	0	0	0	0	539	522	
ORDER	1458	202	5	5	0	0	17	16	
WITHDRAW	16	3035	214	214	0	0	605	332	

IMS Connect response time

IMS transit breakdown and resource usage

. . .	Avg InputQ Time	Avg Process Time	Avg OutputQ Time	Avg Total Time	Avg CPU Time	Avg DB Call Count	. . .
	5	54	0	60	3	2	
	6	7	4203	4216	2	1	
	8	213	17528	17749	1	1	
	8	291	2251	2549	3	1	

Connect Output

. . .	Avg Confirm Time	Avg Output Time	Avg XmitExit Time
	6	0	0
	0	1	0
	179	0	0
	249	1967	0

- **Combined IMS Connect and IMS log metrics**
- **Design a Form to include required information only**
- **No restriction on report page width**
- **Export to CSV or DB2 as an alternative to reporting**

Programming Performance Tidbits

Reduce the number of DLI calls:

- Use path calls.
- Eliminate redundant calls.
- Use single segment input messages.
- Send single segment output messages.

Reduce the number of I/O IWAITS:

- Use fully qualified calls.
- Use XDFLD name in SSA when PCB has PROCSEQ.
- Do not use PROCOPT=GOT, use PROCOPT=GON.

- Use DBB in production instead of DLI

IMS Performance Analyzer
IMS Problem Investigator

Application Program Performance

- Read & Load Applications can benefit from using utility power

Application Programs that are written for sequential reading (often reporting programs) can be given the speed of utilities without any changes to the application

Application Load Programs can also gain the speed of a Load utility without any changes to the application

Unload

- optimized for GN and GNP calls with unqualified segment search arguments (SSAs) calls.

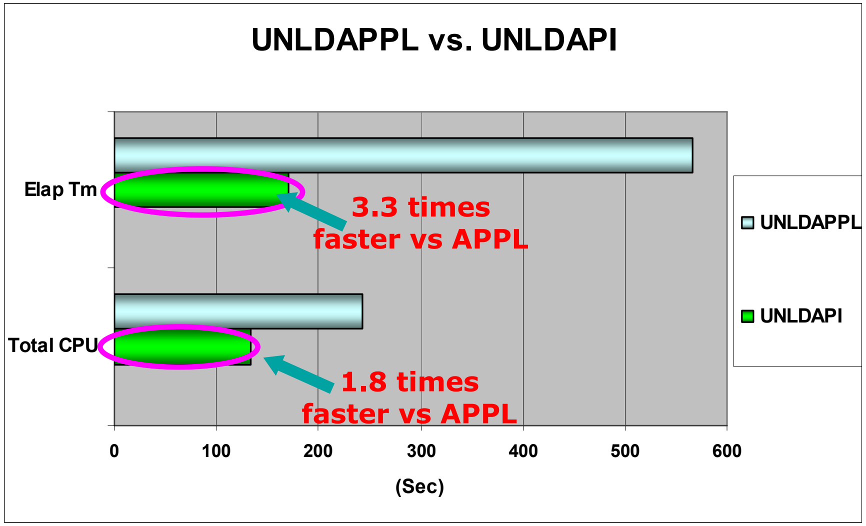
Load

- Requires LOAD PCB
- Including HALDB loads

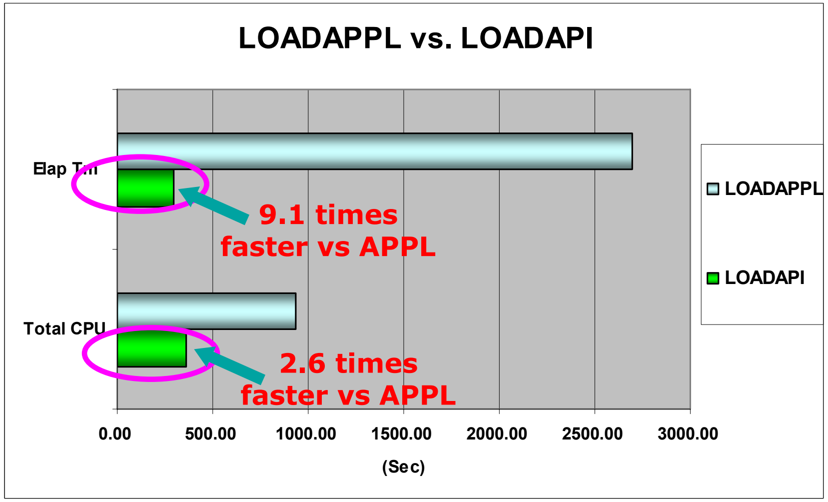
REDUCES Application Elapsed and CPU Times

IMS HP Unload
IMS HP Load

Unload/Load API versus Application Measurements



- Unload Comparison
  - UNLDAPPL
    - Elapsed Time (sec): 567
    - Total CPU: 242.84
  - UNLDAPI
    - Elapsed Time (sec): 171
    - Total CPU: 132.97
- Load Comparison
  - LOADAPPL
    - Elapsed Time (sec): 2,696
    - Total CPU: 931.61
  - LOADAPI
    - Elapsed Time (sec): 296
    - Total CPU: 360.67



- **Faster & less CPU**
  - UNLDAPPL vs. UNLDAPI
    - Elapsed Time: **3.3 times faster**
    - Total CPU: **1.8 times less CPU time**
  - LOADAPPL vs. LOADAPI
    - Elapsed Time (sec): **9.1 times faster**
    - Total CPU: **2.6 times less CPU time**

Logging Performance Tidbits

Performance reflected in DLI-NOT-IWAIT time:

- Make database buffer subpools large
  - avoid LWA due to buffer flush
- Use VSAM background write.
- Define databases as unrecoverable.
- Use 24 K or 28 K blocking.
  - This allows the OLDS buffers to be placed in real storage
- above the 2 GB bar (by being multiples of 4 K).

IMS Performance Analyzer
IMS Buffer Pool Analyzer

Database Performance

- **Access Methods**
- **Block sizes, CI sizes and Record sizes**
- **Free Space**
- Randomization Parameters
- Fixed Length vs. Variable Length
- **Pointer Options**
- SCAN parameter on the DATASET statement
- Multiple data set groups
- **Compression**
- Encryption
- Secondary Indexes
- Fast Path considerations
- Non-Recoverable databases
- **OSAM vs. VSAM**
- **Buffer Life Concept**

Database Access Methods – performance

To choose an IMS access method:

- What type of processing is done (Choices are shown in preferred order)?
  - Direct: Use DEDB, HDAM, HIDAM, or HISAM.
  - Sequential: Use DEDB (Seq Rand), HDAM (Seq Rand), HIDAM, or HISAM.
  - Both: Use DEDB (Seq Rand), HDAM (Seq RAND), or HIDAM.
- Is the data volatile? Yes, use DEDB, HDAM, or HIDAM.
- Do the database records vary in length? Yes, use DEDB, HDAM, or HIDAM.
- Are logical relationships needed? Yes, use HDAM or HIDAM.
- Are secondary indexes needed? Yes, use HDAM or HIDAM.
- Is there a need for a journaling capability? Yes, use DEDB.

Note: Wherever HDAM or HIDAM is shown, partitioning (HALDB) is preferred.
Seq Rand means using a Randomizer that maintains the key sequence.

Block or CI Size

Larger CIs or blocks:

- Improve sequential processing.
- Reduce the number of IWAITS.
- Increase IWAIT time per IWAIT.
- Decrease total IWAIT time.

Smaller CIs or blocks might:

- Improve random processing.
- Increase number of IWAITS.
- Reduce IWAIT time per IWAIT.

IMS Performance Analyzer
IMS Problem Investigator

Freespace Tidbits

Reduce number of reorgs by getting this right

- Purpose: Minimize CI/CA split activity:
  - No additions:
    - No need for FREESPACE.
  - Few additions:
    - No FREESPACE or some FREESPACE in the CI.
  - Evenly distributed additions:
    - FREESPACE in the CA or FREESPACE in both CI and CA.
  - Unevenly distributed additions:
    - Specify a small amount of FREESPACE.
- FREESPACE cost:
  - Additional DASD space which might remain unused.
  - Additional I/O to sequentially process the same number of records.
  - Additional number of index records (levels).

IMS HP Pointer Checker
IMS Performance Analyzer

Pointer TidBits

- Use child and twin pointers instead of hierarchic pointers.
- Do not specify twin backward pointers for dependent segments unless you satisfy the criteria for deletes with logical relationships.
- Never specify twin forward only pointers for HIDAM roots.
- Never specify twin forward and backward pointers for HDAM roots.
- Specify no twin pointers for HIDAM and PHIDAM roots.
- If you specify RULES=(,LAST) or use last as the default for segments without sequence fields, you should define a physical child last pointer from the parent if there might be a long twin chain.

Compression Tidbits

The considerations are:

- Improves DASD space utilization (more data in block)
- Improves buffer space utilization
- Might reduce I/O
- Increases CPU time unless you are using Hardware Data Compression

IMS HP Pointer Checker
IMS Hardware Compression Ext

Database – OSAM vs VSAM

- Tests were run in a controlled environment in the Silicon Valley Laboratory using 10 HIDAM databases.
- The first set of tests were run with the databases defined with VSAM, and then a second set of tests were run with OSAM using the same workload that was used in the first test.
- Set one
  - three BMPs each executing 2 000 000 total database calls.
  - There were 10 qualified GHU calls performed along with 1 000 000 qualified GHN calls and 1 000 000 replace calls.
- Set two
  - four BMPs each executing 4 500 000 total database calls.
  - There was one qualified GHU call performed along with 1 000 qualified GHN calls, 1 000 replace calls, and 4 000 000 GN calls

IMS Performance Analyzer

Database – OSAM vs VSAM

Type	Avg CPU Time	Elapsed Time	Delta CPU	Delta Elapsed
BMP Set One				
VSAM	168	8.71		
OSAM	136	6.01	19.04% reduction	27.59 % reduction
OSAM SB	138	6.93	18.8% reduction	27.34% reduction

BMP Set Two				
VSAM	98	5.45		
OSAM	57	3.50	41.83% reduction	35.78% reduction
OSAM SB	61	1.16	37.75% reduction	78.59% reduction

OSAM vs VSAM ---- Why??

- OSAM writing of multiple blocks
  - Sorts by physical location
  - Chained writes in parallel
- Shorter processor instruction path length
- OSAM sequential buffering
- OSAM data sets up to 8 Gb
- Reuse OSAM data sets

Application Buffers – cheap performance tuning

- Changes to buffers can be done without a reorg and without an application change

- Easy BUT the big question is HOW and HOW MUCH?
- Criteria is BUFFER LIFE CONCEPT
- User Think Time

Questions:

- If I make a change, will there be any improvement?
- Should the amounts go up or down?
- What would happen if I moved a database to its own subpool?

IMS Performance Analyzer
IMS Buffer Pool Analyzer

IMS Buffer Pool Analyzer

PAGE 2 IMS BUFFER POOL ANALYZER 1.1.0
DATABASE SUBPOOL SUMMARY

BUFFER SIZE	TYPE	SUB-POOL	VSAM TYPE	NUMBER BUFFER	BUFFER LIFE	HIT RATIO	REQUESTS PER SECOND	READS PER SECOND
4,096	OSAM	NONE		300	12.65	90.1%	240.7	23.7
4,096	OSAM	LDB		300	12.51	97.6%	1,011.1	23.9
6,144	OSAM	NONE		200	35.61	95.9%	137.6	5.6
8,192	OSAM	NONE		200	114.25	88.2%	14.9	1.7
8,192	OSAM	PX01		500	10.95	85.3%	310.7	45.6
8,192	OSAM	PX16		500	13.30	85.7%	263.7	37.5
4,096	VSAM	XXXX	BOTH	1,000	8.02	88.2%	574.9	67.6

IMS Buffer Pool Analyzer

NUMBER BUFFERS	POOL SIZE (K)	PROJECTIONS HIT RATIO	READS PER SECOND	HITS PER SECOND	BUFFER LIFE	MARGINAL REDUCTION
4	16	81.3%	188.6	822.5	0.02	
8	32	85.3%	148.5	862.6	0.05	9,020
16	64	88.4%	117.2	893.9	0.13	3,517
32	128	91.2%	88.9	922.2	0.35	1,593
64	256	93.1%	69.0	942.0	0.92	557
128	512	94.9%	51.1	960.0	2.50	252
192	768	95.9%	40.5	970.6	4.73	149
256	1,024	96.7%	33.0	978.1	7.74	104
384	1,536	97.4%	26.1	985.0	14.69	48
512	2,048	97.6%	24.0	987.0	21.24	14
768	3,072	97.6%	23.4	987.7	32.73	2
1,024	4,096	97.6%	23.4	987.7	43.64	0
1,536	6,144	97.6%	23.4	987.7	65.46	0
2,048	8,192	97.6%	23.4	987.7	87.28	0

When to Reorganize

- Classic Reasons:

1. Extents
2. Freespace Statistics
3. CI/CA Splits
4. HDAM - % of roots in home block.

- But does that tell the true story:
  - Is the area of the database that is “out of order” accessed by applications?
  - Is it affecting application performance?
  - Gather stats post-reorg so you can tell if a reorg will make a difference
    - Application performance deteriorating
    - Too many physical I/Os to DASD

IMS HP Pointer Checker
IMS Performance Analyzer

Where to go for more Performance Hints

Redbook:

- IMS Performance and Tuning Guide
  - December 2006
  - SG24-7324-00

Technical Trivia

BIT = BInary digiT a 0 or 1
Nibble = 4 BITs
Byte = 8 BITs
Kilobyte = 8,192 BITs
Megabyte = 8,388,608 BITs
Gigabyte = 8,589,934,592 BITs
Terabyte = 8,796,093,022,208 BITs
Petabyte = 9,007,199,254,740,992 BITs
Exabyte = 9,223,372,036,854,775,808 BITs
Zettabyte = 9,444,732,965,739,290,427,392 BITs
Yottabyte = 9,671,406,556,917,033,397,649,408 BITs
Xonabyte = 9,903,520,314,283,042,199,192,993,792 BITs
Wekabyte = 10,141,204,801,825,835,211,973,625,643,008 BITs
Vundabyte = 10,384,593,717,069,655,257,060,992,658,440,192 BITs

Kilobyte = 1,024 bytes or 2^{10}
Megabyte = 1,048,576 bytes or 2^{20}
Gigabyte = 1,073,741,824 bytes or 2^{30}
Terabyte = 1,099,511,627,776 bytes or 2^{40}
Petabyte = 1,125,899,906,842,624 bytes or 2^{50}
Exabyte = 1,152,921,504,606,846,976 bytes or 2^{60}
Zettabyte = 1,180,591,620,717,411,303,424 bytes or 2^{70}
Yottabyte = 1,208,925,819,614,629,174,706,176 bytes or 2^{80}
Xonabyte = 1,237,940,039,285,380,274,899,124,224 bytes or 2^{90}
Wekabyte = 1,267,650,600,228,229,401,496,703,205,376 bytes or 2^{100}
Vundabyte = 1,298,074,214,633,706,907,132,624,082,305,024 bytes or 2^{110}

What would you like to see on these calls?

- Janet LeBlanc
  - leblancj@ca.ibm.com
- Sally Touscany
  - touscany@us.ibm.com
  - 412-667-3334

Thank You for Joining Us today!

Go to **www.ibm.com/software/systemz** to:

- Replay this teleconference
- Replay previously broadcast teleconferences
- Register for upcoming events